En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

(57) Abrégé : Ce procédé de recherche d'informations dans des documents stockés dans une mémoire électronique comporte les étapes suivantes:- sélection d'au moins un document parmi les documents stockés, à partir d'une requête comportant au moins une chaîne de caractères prédéterminée, puis - extraction d'un résultat en vue de son affichage sous forme d'un aperçu d'informations relatives au document sélectionné, et - préalablement aux étapes de sélection et d'extraction, génération d'une table de représentation des documents stockés, comportant une chaîne de caractères comprenant au moins une partie des informations des documents stockés. Lors de l'étape d'extraction, on génère le résultat à l'aide de la table de représentation, à partir d'informations contenues dans la chaîne de caractères de la table de représentation jugées pertinentes en fonction de la requête.

Procédé de recherche d'informations, moteur de recherche et microprocesseur pour la mise en œuvre de ce procédé

La présente invention concerne un procédé de recherche d'informations dans des documents stockés dans une mémoire électronique. L'invention concerne également un microprocesseur pour la mise en œuvre de ce procédé et un moteur de recherche.

Plus précisément l'invention concerne un procédé de recherche d'informations du type comportant les étapes suivantes :

- sélection d'au moins un document parmi les documents stockés, à partir d'une requête comportant au moins une chaîne de caractères prédéterminée, puis
- extraction d'un résultat en vue de son affichage sous forme d'un aperçu d'informations relatives au document sélectionné, et
- préalablement aux étapes de sélection et d'extraction, génération d'une table de représentation des documents stockés, comportant une chaîne de caractères comprenant au moins une partie des informations des documents stockés.

Un tel procédé est connu. En effet, devant la multiplication des documents sous forme de fichiers obtenus par traitement de texte ou de courriers électroniques disponibles dans les micro-ordinateurs et les réseaux internes des entreprises, la nécessité de disposer d'un procédé de recherche d'informations permettant de retrouver rapidement un document par un élément de son contenu s'impose de plus en plus. De nouveaux logiciels permettent d'ores et déjà de rechercher l'information sous forme de texte dans tout type de document, y compris dans les pièces jointes de courriers électroniques. Pour cela, préalablement à toute recherche d'informations, une table de représentation des documents stockés, généralement appelée index, permet de reprendre pour chacun des documents stockés, une liste de mot-clés représentatifs de ce document et à partir desquels le document peut être éventuellement sélectionné sur la base d'une requête.

Toutefois, malgré cela, les temps de recherche sont encore importants, car lorsqu'un document a été sélectionné, il est souvent nécessaire d'ouvrir le document avec le programme de visualisation qui lui est associé pour s'assurer qu'il s'agit bien d'un document recherché. Plus grave encore, lorsque l'on a ouvert une dizaine de documents (traitement de texte, tableaux, courriers électroniques, etc.), il devient difficile de passer de l'un à l'autre.

L'invention vise à remédier à ces inconvénients en fournissant un procédé de recherche d'informations permettant à un utilisateur de visualiser de façon rapide et

efficace le contenu de documents sélectionnés en réponse à une requête qu'il a formulée.

L'invention a donc pour objet un procédé de recherche d'informations du type précité, caractérisé en ce que, lors de l'étape d'extraction, on génère le résultat à l'aide de la table de représentation, à partir d'informations contenues dans la chaîne de caractères de la table de représentation jugées pertinentes en fonction de la requête.

Ainsi, pour visualiser le contenu des documents sélectionnés, il n'est pas nécessaire d'ouvrir ces derniers, puisque le contenu pertinent est directement extrait d'une même table de représentation pour l'ensemble des documents.

De préférence, lors de l'étape de sélection, on compare la chaîne de caractères prédéterminée de la requête à la chaîne de caractères de la table de représentation, notamment par balayage séquentiel de la table de représentation, pour sélectionner au moins un document parmi les documents stockés.

Ainsi, la table de représentation est également utilisée en tant que table d'indexation des documents stockés. Elle est donc utilisée à la fois pour la visualisation du contenu des documents stockés et pour la recherche des ces documents à partir d'une requête comportant au moins une chaîne de caractères prédéterminée. Le balayage séquentiel de la chaîne de caractères contenue dans la table de représentation permet d'augmenter sensiblement l'efficacité de la recherche.

De façon optionnelle, au moins un document stocké étant de type courrier électronique et comportant plusieurs rubriques distinctes choisies parmi l'ensemble d'éléments constitué d'une adresse d'un émetteur, d'une adresse d'un destinataire, d'un en-tête, d'un corps de message, et d'au moins une pièce jointe, la chaîne de caractères de la table de représentation comporte au moins une partie des informations de type texte de chaque rubrique du document de type courrier électronique.

Ainsi, on peut effectuer une recherche dans un ensemble de courriers électroniques stockés en tenant compte non seulement du contenu de ces courriers électroniques mais également éventuellement du contenu de pièces jointes à ces courriers électroniques ou d'autres parties de ces courriers électroniques, telles que les en-tête.

Dans ce cas, pour le document de type courrier électronique, on peut balayer séquentiellement les informations concernant la pièce jointe avant les informations concernant toute autre rubrique de ce document.

En effet, il arrive fréquemment que les pièces jointes des courriers électroniques comportent les informations les plus pertinentes.

De façon optionnelle, la chaîne de caractères de la table de représentation comporte en outre pour chaque document stocké des informations d'identification de ce document.

Ainsi, la visualisation et la recherche d'informations peuvent tenir compte de ces informations d'identification.

De façon optionnelle, on stocke en mémoire au moins une partie du résultat de la recherche d'informations.

De façon optionnelle également, la partie du résultat de la recherche d'information stockée en mémoire est stockée dans un fichier apte à comporter plusieurs résultats de plusieurs recherches.

Dans un mode de réalisation possible, lors de l'étape d'extraction du résultat, le procédé de recherche d'informations comporte les étapes suivantes :

- extraction des informations contenues dans la chaîne de caractères de la table de représentation jugées pertinentes en fonction de la requête,
- transmission de ces informations vers un terminal distant par l'intermédiaire d'un réseau de transmission de données,

et l'affichage du résultat est réalisé par le terminal distant.

Lors de l'étape de génération de la table de représentation des documents stockés, on peut effectuer une conversion pour que tout caractère affichable d'une zone de type texte des documents stockés soit codé :

- soit sur un octet ;
- soit à l'aide d'une balise insérée dans la table de représentation et suivie d'un code sur un octet

Dans un mode de réalisation particulier de l'invention, lors de l'étape de génération de la table de représentation, on insère dans la chaîne de caractères de la table de représentation au moins un ensemble de données délimité par au moins une balise pour compléter les informations comprises dans cette chaîne de caractères.

On peut ainsi imaginer insérer des données supplémentaires, à l'aide de balises prédéfinies pour améliorer la visualisation de documents sélectionnés ou pour augmenter la performance de la recherche d'informations. L'insertion de ces données supplémentaires à l'aide de balises directement dans la chaîne de

caractères de la table de représentation permet de ne pas réduire les performances de la recherche d'informations.

Ainsi, par exemple, l'ensemble de données comporte des données d'aide à la présentation de l'aperçu, utilisées lors de l'étape d'extraction du résultat.

Les données supplémentaires sont par exemple des informations de mise en page permettant d'améliorer la visualisation du contenu des documents sélectionnés, notamment pour rester fidèle à la mise en page du contenu tel qu'il était présenté dans le document lui-même.

L'ensemble de données peut également comporter des données d'aide à la sélection d'au moins un document.

On peut ainsi imaginer des données supplémentaires insérées à l'aide de balises d'accentuations, de synonymies, d'écriture phonétique, etc. Ainsi, ces données d'aide à la sélection permettent de sélectionner des documents comportant au moins une chaîne de caractères voisine de la chaîne de caractères prédéterminée définie dans la requête.

Un procédé de recherche d'informations selon l'invention peut en outre comporter l'une ou plusieurs des caractéristiques suivantes :

- chaque balise insérée dans la chaîne de caractères de la table de représentation comporte au moins un caractère d'échappement codé sur un octet n'appartenant pas aux caractères affichables figurant dans les 128 premières positions de la table de codification ASCII,
- on insère dans la chaîne de caractères de la table de représentation au moins une zone d'informations de type numérique codée sur un nombre prédéterminé d'octets délimité par au moins une balise d'indication de cette zone numérique,
- la balise d'indication de la zone numérique est en outre une balise d'indication d'une convention de présentation de cette zone numérique,
- les documents stockés étant répartis en différents types de documents, on définit pour chaque type de documents un ensemble de balises destinées à être insérées dans la chaîne de caractères de la table de représentation, chaque balise de cet ensemble ayant une signification spécifique à ce type de documents,
- on insère dans la chaîne de caractères de la table de représentation au moins un ensemble de données exprimées en écriture phonétique délimité par au moins une balise d'indication d'écriture phonétique,

- on insère dans la chaîne de caractères de la table de représentation au moins une balise d'indication qu'un nombre prédéterminé de caractères suivant cette balise dans la chaîne de caractères de la table de représentation n'a pas à être examiné lors de l'étape de sélection,
- on insère dans la chaîne de caractères de la table de représentation au moins un ensemble de données correspondant à une analyse grammaticale d'une partie du contenu d'au moins un document stocké, délimité par au moins une balise d'indication d'analyse grammaticale,
- on insère dans la chaîne de caractères de la table de représentation au moins un ensemble de données correspondant à des méta-données de description d'une partie du contenu d'au moins un document stocké, délimité par au moins une balise d'indication de méta-données,
- on insère dans la chaîne de caractères de la table de représentation au moins une balise pour lancer un programme prédéterminé.

En outre, un procédé de recherche d'informations selon l'invention peut comporter la caractéristique selon laquelle :

- chaque document stocké comportant des informations réparties dans plusieurs rubriques distinctes prédéterminées communes à tous les documents stockés, le résultat est affiché sous la forme d'un aperçu comportant une zone d'aperçu pour chaque rubrique distincte commune et comportant une liste de documents initialement sélectionnés pour des informations qu'il contiennent jugées pertinentes en fonction de la recherche,
- chaque zone d'aperçu est désactivable, et
- lorsqu'on désactive au moins une zone d'aperçu, on maintient uniquement dans la liste affichée chaque document initialement sélectionné pour des informations jugées pertinentes que ce document comporte dans au moins une rubrique correspondant à au moins une zone d'aperçu qui reste activée.

A l'aide de ces caractéristiques supplémentaires, le procédé de recherche d'informations permet à l'utilisateur d'effectuer un choix rapide dans un ensemble de documents sélectionnés fournis en réponse à sa requête.

L'invention concerne également un moteur de recherche d'informations dans des documents stockés dans une mémoire électronique, comportant :

- des moyens de génération d'une table de représentation des documents stockés, cette table comportant une chaîne de caractères comprenant au moins une partie des informations des documents stockés,
- des moyens de sélection d'au moins un document parmi les documents stockés, à partir d'une requête comportant au moins une chaîne de caractères prédéterminée,

caractérisé en ce qu'il comporte des moyens d'extraction d'un résultat à l'aide de la table de représentation, à partir d'informations contenues dans la chaîne de caractères de la table de représentation jugées pertinentes en fonction de la requête, en vue de l'affichage de ce résultat sous forme d'un aperçu d'informations relatives au document sélectionné.

Enfin, l'invention concerne également un microprocesseur comportant des instructions programmées pour la mise en œuvre d'un procédé de recherche d'informations tel que défini précédemment.

Un microprocesseur selon l'invention peut en outre comporter des moyens de stockage d'au moins une table dictionnaire comprenant un ensemble de mots dans une langue prédéterminée, chaque mot étant associé dans cette table dictionnaire à des données d'analyse grammaticale.

L'invention sera mieux comprise à l'aide de la description qui va suivre, donnée uniquement à titre d'exemple et faite en se référant aux dessins annexés dans lesquels :

- la figure 1 représente schématiquement les étapes successives mises en œuvre pour la génération d'une table de représentation de documents stockés, dans un procédé de recherche d'informations selon l'invention ;
- la figure 2 représente schématiquement un exemple de chaîne de caractères contenue dans la table de représentation de la figure 1 ;
- les figures 3 et 4 représentent des fenêtres de visualisation d'une sélection de documents, affichées lors de la mise en œuvre d'un mode de réalisation particulier de l'invention ; et
- la figure 5 représente schématiquement un dispositif comportant un microprocesseur maître et plusieurs coprocesseurs pour l'exécution rapide d'un procédé selon l'invention.

Comme cela est représenté sur la figure 1, un procédé selon l'invention utilise les éléments suivants :

- un ensemble de documents sur lesquels on est appelé à effectuer des recherches, à savoir tous types de documents comportant du texte tels que des documents issus de traitements de texte, tableurs (notés Doc), ou des courriers électroniques (notés Mail) avec éventuellement leurs pièces jointes (notées Att, Zip), ces documents étant stockés soit sur un ordinateur à partir duquel sont exécutées les recherches, soit dans des réseaux internes d'entreprises, soit en dehors et accessibles via Internet,
- un ensemble de tables, dites tables d'index, pour effectuer les recherches, et
- un ensemble de tables de représentation des documents stockés, dites tables des aperçus, pour permettre un affichage rapide des résultats.

Dans un mode préféré de l'invention, ce sont les mêmes tables qui sont utilisées à la fois pour effectuer la recherche et afficher les aperçus, c'est-à-dire que ce sont les tables d'index qui sont utilisées en tant que tables de représentation des documents stockés pour afficher les aperçus. Par la suite ces tables seront appelées tables d'index et d'aperçu (notées TIA).

Un procédé de recherche selon l'invention nécessite les étapes suivantes :

- génération d'une table d'index et d'aperçu (ie. une table de représentation des documents stockés) comportant au moins une partie des informations des documents stockés,
- recherche de documents par la sélection d'au moins un document parmi les documents stockés, à partir d'une requête comportant au moins une chaîne de caractères prédéterminée,
- affichage d'un résultat sous forme d'un aperçu d'informations relatives au(x) document(s) sélectionné(s).

Génération de la table d'index et d'aperçu.

La table d'index et d'aperçu doit permettre une recherche rapide et un affichage rapide des aperçus. Elle contient pour chaque document les deux types d'informations suivantes :

- d'une part, le contenu intégral ou partiel du document en format texte, non compressé, c'est-à-dire tout élément qui peut être affiché sous

forme de texte (dans le cas des courriers électroniques le contenu des documents attachés, qu'il soit sous forme compressée ou non, est également mémorisé dans la table d'index et d'aperçu).

- d'autre part, des éléments d'identification du document tels que le nom du document, son objet, une date, sa longueur, des mots clefs, un chemin d'accès au document sur le disque, etc. (pour les courriers électroniques, le nom de l'émetteur sous forme d'adresse électronique et sous forme d'alias, le nom des destinataires, des copies, un nom de dossier, etc.).

Tous les documents sont stockés les uns à la suite des autres soit dans une table d'index et d'aperçu unique, soit dans plusieurs tables d'index et d'aperçu, une par type de document par exemple (notées TIA-Doc TIA-Mail). Comme représenté sur la figure 2, chaque document tel que Tia-doc est représenté par un en-tête (noté Tia-Id) suivi de tous les champs en format texte (noté Tia-txt) susceptibles d'être sélectionnés lors d'une recherche d'informations.

Dans un mode de réalisation préféré de l'invention, on utilise un système de séparateurs entre les différents documents, et entre les différents éléments à l'intérieur de chaque document afin de permettre un balayage rapide de la table d'index et d'aperçu.

L'en-tête Tia-Id regroupe des données de type numérique, ainsi que des textes sur lesquels on n'effectue pas de recherche :

- un caractère séparateur '0xff' ou tout autre caractère qui ne peut pas figurer dans un fichier texte, situé au début de l'en-tête,
- la longueur de l'en-tête,
- des données numériques telles que des longueur de blocs, des compteurs divers,
- des données numériques susceptibles d'être recherchées, appelées par la suite rubriques, telles que la longueur ou la date du document,
- données alphabétiques qui ne font pas partie du champs des recherches (nom de machine, client, langue, tables de conversion, etc.).

A la suite on trouve une partie texte (notée Tia-txt), comportant tous les éléments sur lesquels sont effectués les recherches en format texte. Il s'agit des contenus, des mots-clefs, des éléments d'identification des documents. Ces

différents éléments, appelés par la suite rubriques, sont stockés les uns à la suite des autres sous forme de texte, et ils sont séparés par des caractères séparateurs.

Dans un mode de réalisation préféré de l'invention, le contenu de chacune des pièces jointes des courriers électroniques est mémorisé dans une table d'index et d'aperçu séparée (notée TIA-Att) dite table d'index des pièces jointes et un document donné n'y figure qu'une seule fois, même s'il appartient à plusieurs courriers électroniques ou à plusieurs fichiers compressés Zip eux-mêmes attachés en pièce jointe.

Les tables d'index et d'aperçu sont générées puis régulièrement mises à jour grâce à des convertisseurs (notés Conv) qui, à partir des documents de départ (traitement de texte, tableurs, présentations, courriers électroniques ...) extraient tous les éléments utiles pour la consultation de ces tables au moment de la recherche d'information, puis par la suite pour l'affichage des résultats sous forme d'aperçu.

Recherche de documents.

Hormis les logiciels de recherche documentaire ou moteurs de recherche sur Internet qui sont très rapides puisqu'ils utilisent un thésaurus, en général, les logiciels de recherche sur ordinateur commencent par balayer une table d'index des fichiers sur le disque dur de l'ordinateur, communément appelée FAT, ou une table équivalente qui permet de vérifier si le nom du fichier, le type du fichier, sa longueur ou sa date satisfont aux critères de recherche. Si c'est le cas, et dans le cas où l'on doit effectuer la recherche sur des mots contenus dans les documents eux-mêmes, on balaie alors séquentiellement le contenu de chacun des documents qui correspondent à ces premiers critères de recherche, pour vérifier si les mots recherchés figurent dans ce document. Il s'avère que cette technique, consistant à explorer d'abord une table d'index, puis si nécessaire, une seconde table contenant les textes eux-mêmes, est beaucoup plus lente que celle qui consiste à balayer séquentiellement une table d'index et d'aperçu qui contient tous les contenus des documents ainsi qu'il est décrit ci-après.

Pour effectuer une recherche sur un ou plusieurs mots ou parties de mot, on balaie séquentiellement la table d'index et d'aperçu comme suit :

- quand on rencontre un séparateur de document (égal à 0xff), on analyse les éléments de l'en-tête Tia-id du document qui suit puis on se positionne sur le premier caractère de la zone Tia-txt correspondant

aux éléments sur lesquels on veut effectuer la recherche en format texte dans ce document,

- ensuite, on balaie la zone Tia-txt pour vérifier si elle contient une partie ou la totalité des mots recherchés. Si ce n'est pas le cas, on passe au document suivant, sinon le décompte du nombre de séparateurs permet de savoir de quelle rubrique il s'agit, et grâce aux données de l'en-tête précédemment chargé, on dispose alors de tous les éléments nécessaires pour afficher le résultat de la recherche.

Dans un mode de réalisation préféré de l'invention, on commence par balayer la table d'index des pièces jointes TIA-Att et, chaque fois qu'une pièce jointe comporte le ou les mots recherchés, on mémorise temporairement dans une table un identifiant de cette pièce jointe, ce qui permet, par la suite, lors d'un balayage de la table des courriers électroniques TIA-Mail d'identifier les courriers qui ont des pièces jointes contenant les mots recherchés.

Dans le cas où l'on recherche des informations dans des documents, à partir d'une requête comportant deux chaînes de caractères prédéterminées, on peut procéder de deux manières différentes :

- sans duplication de documents : au cours d'une première phase on lance la recherche par balayage sur la totalité de la table de représentation, et on mémorise les adresses des documents qui contiennent la première des deux chaînes de caractères prédéterminées, puis au cours d'une deuxième phase, on lance la recherche par balayage des seuls documents dont on a gardé l'adresse, pour sélectionner ceux qui contiennent la seconde chaîne de caractères prédéterminée ; ou
- avec duplication de documents dans une nouvelle table secondaire dite « table secondaire de représentation » : au cours d'une première phase on lance la recherche par balayage sur la totalité de la table de représentation, et par duplication, on crée un nouvelle table secondaire de représentation à partir des documents qui contiennent la première des deux chaînes de caractères prédéterminées, puis au cours d'une deuxième phase, on lance la recherche par balayage sur la nouvelle table secondaire de représentation que l'on vient de créer de façon à sélectionner les documents qui contiennent aussi la seconde chaîne de caractères prédéterminée.

Affichage d'un résultat.

Les informations relatives aux documents sélectionnés à l'issue de la recherche sont affichées sous la forme d'un tableau dit tableau des documents trouvés, comportant une ou plusieurs lignes pour chaque document trouvé et plusieurs colonnes correspondant chacune à une ou plusieurs desdites rubriques.

Quand une ligne du tableau est sélectionnée, par exemple un courrier électronique, le contenu Tia-txt de ce courrier est extrait de la table d'index et d'aperçu TIA puis affiché dans une fenêtre séparée dite fenêtre des aperçus. Quand on passe à la ligne suivante du tableau, c'est le contenu de ce nouveau courrier qui est affiché dans la fenêtre des aperçus. Quand un courrier électronique Mail contient une ou plusieurs pièces jointes Att, le nom des pièces jointes est affiché à l'écran, et quand on sélectionne l'une d'elle, son contenu Tia-Att est extrait de la table des pièces jointes TIA-Att puis affiché dans la fenêtre des aperçus, sans qu'il soit nécessaire d'exécuter un logiciel de présentation d'informations (traitement de texte, tableur, ...) qui lui est associé.

Cette opération est extrêmement rapide puisque le contenu affiché fait partie de la table qui est explorée au cours de l'étape de recherche.

Le fait de lancer au moins une recherche, puis de sélectionner les seuls documents utiles en vue de traiter un problème, représente une opération à la fois coûteuse en temps et en compétence, c'est-à-dire qu'une telle sélection apporte de la valeur ajoutée par rapport à l'information brute de départ. Avec les techniques actuelles de courrier électronique, si l'on désire transmettre cette information à une autre personne, tous les documents vont être transmis sous forme de pièces jointes à un courrier, et le destinataire sera amené à refaire une partie du travail de sélection qui a déjà été réalisée.

C'est pourquoi il est préférable de lui transmettre un dossier appelé par la suite « fichier-conteneur » (noté File-Cont) qui contient non seulement les documents de départ (traitements de texte, tableurs, courriers électroniques, ...), mais également tous les éléments qui vont lui permettre de récupérer tout le travail de classement qui avait été ajouté par l'auteur de la recherche initiale.

Pour cela, il suffit de disposer d'un fichier-conteneur vers lequel, on peut avec une fonction « copier-coller », copier une ou plusieurs lignes du tableau des documents trouvés. Grâce à cette opération, on mémorise dans une mémoire permanente, toutes les informations relatives à chaque ligne du tableau, à savoir, le

contenu du document original avec sa mise en page, les dessins, images, sons, animations, etc, le texte Tia-txt nécessaire pour afficher l'aperçu, et toutes les informations que l'utilisateur de départ aura ajoutées à ces informations de départ pour en rendre la lecture plus rapide, et la présentation plus pertinente (par exemple les critères de recherche, les modes de tri par colonnes, ou bien la façon d'ordonner les lignes du tableau des documents trouvés, les statistiques sur la recherche ...).

Ce fichier-conteneur, à l'instar d'une chemise de courrier, peut être transmis à une autre personne soit sous forme de fichier via un réseau interne d'entreprise, soit sous forme de pièce jointe attachée à un courrier électronique. Le destinataire pourra voir le contenu de ce fichier-conteneur, affiché sous forme de tableau, de manière analogue au tableau des documents trouvés, chaque ligne du fichier-conteneur correspondant à une ligne du tableau des documents trouvés. De la même manière, grâce à la fenêtre pour l'affichage de l'aperçu, il est possible aussi de voir rapidement le contenu des documents contenus dans le fichier-conteneur (courriers électroniques, traitement de texte, tableurs ...) sans avoir besoin d'ouvrir les documents avec les logiciels de présentation d'informations qui leur sont associés.

Le fichier-conteneur peut à son tour être modifié ou enrichi avec d'autres documents, puis transmis à d'autres destinataires. Lorsqu'il est utilisé en tant que pièce jointe attachée à un courrier électronique, il peut, à son tour, être exploré par le moteur de recherche, et les résultats de la recherche peuvent être insérés dans un nouveau fichier–conteneur.

Les informations relatives aux documents trouvés à l'issue de la recherche sont affichées sous la forme d'un aperçu comportant une zone d'aperçu pour chaque rubrique et comportant une liste de documents initialement sélectionnés pour des informations qu'ils contiennent jugées pertinentes en fonction de la recherche.

Plus précisément, elles sont affichées par exemple sous la forme d'un tableau comportant une ou plusieurs lignes pour chaque document sélectionné et plusieurs colonnes correspondant chacune à une ou plusieurs desdites rubriques.

La Figure 3 montre un exemple de résultat de recherche dans des courriers électroniques dans lequel les lignes L1, L2, L3 et L4 contiennent une séquence de caractères recherchée « Paris ».

Le titre de chaque colonne comporte à la fois l'intitulé de la rubrique correspondante, ainsi qu'une case à cocher ou un dispositif équivalent fonctionnant comme suit :

- si la case est cochée, la colonne est activée et toutes les lignes qui comportent le ou les mots recherchés dans la rubrique correspondant à cette colonne, sont affichées,
- dans le cas contraire, sont masquées les lignes qui contiennent le ou les mots recherchés qui figurent uniquement dans la rubrique correspondant à la colonne.

Dans l'exemple de la figure 3, parmi les lignes qui contiennent les informations pertinentes, à savoir la séquence « Paris », on affiche seulement les lignes qui comportent la séquence recherchée dans au moins une des colonnes activées, ce qui est différent du dispositif classique d'onglet consistant à afficher seulement les lignes qui comportent une séquence recherchée dans une rubrique donnée.

De la sorte, simplement en cochant ou décochant une colonne, il est possible de n'afficher qu'une partie des lignes correspondant au résultat de la recherche.

Dans la figure 4, la colonne C3 est désactivée pour masquer tous les courriers pour lesquels « paris » était simplement en copie : la ligne L2 n'apparaît plus, par contre la ligne L3 est toujours affichée car « paris » apparaît dans la colonne C2 de la ligne L3.

Néanmoins, le procédé décrit précédemment peut être encore amélioré pour répondre à plusieurs problèmes.

L'affichage dans la fenêtre d'aperçu ne fait apparaître que le texte brut d'un document sélectionné, exactement comme les courriers électroniques en format brut, c'est-à-dire sans ses éléments de mise en page, ni couleur, ni mots soulignés ou affichés en gras, alors qu'il peut être souhaitable d'afficher ces aperçus avec une présentation améliorée, proche ou équivalente à la présentation initiale du document sélectionné,

Par ailleurs, ce procédé ne donne pas toute satisfaction quand on fait des recherches sur des mots avec des accents : en effet si on cherche le mot «amélioré», les documents contenant seulement «améliore» ne seront pas détectés,

Dans certains cas, on voudrait également, trouver des documents à partir d'un synonyme, ou d'une notion équivalente, par exemple «financer» au lieu de « financement »,

Dans d'autres cas encore, quand il s'agit de montants, on voudrait pouvoir trouver un document qui contient «1.000» quand on recherche «1000», ou l'inverse,

ceci quelle que soit la convention d'écriture (les anglo-saxons utilisent le point à la place de la virgule). De manière analogue, on voudrait faire facilement la différence entre le nombre 1000, et un nombre qui contient les mêmes chiffres comme 10001, ou entre un nombre qui correspond à un montant ou un code article ou un numéro de compte.

Dans d'autres cas enfin, on voudrait pouvoir reconstituer le document texte de départ à partir de la table de représentation des documents stockés, par exemple reconstituer un document généré en format «.rtf» ou un courrier électronique en format «.html», de façon à réduire la place occupée sur disque, ou pour ne travailler que sur une information unique au lieu d'une réplication ajoutée à une information originale, ce qui est beaucoup plus simple et sûr pour tous les traitements informatiques.

D'une manière générale, il est utile d'avoir dans la table de représentation des documents stockés, sous une forme ou sous une autre :

- tous les éléments pour reconstituer l'information de départ,
- les éléments permettant de supporter les approximations dues à l'orthographe, aux accents, aux symboles monétaires, aux notions d'arrondis, et permettant d'utiliser des techniques connues d'analyse automatique de documents,
- les éléments relatifs à la nature d'une information (montant, compteurs, numéro de compte, code d'article, notion de pointeur vers un élément parent ou enfant, etc.) pour pouvoir utiliser ce genre de table dans des applications sans rapport avec la recherche documentaire.

Pour un certain nombre d'informations complémentaires, la meilleure solution consiste à ajouter toute une série de champs à coté du texte brut.

Par contre, pour d'autres il est préférable d'utiliser un système de codification dans lequel les informations sont intimement liées au texte lui-même, grâce à un système de balises analogue à celui que l'on trouve dans des codages comme les formats « .html » ou « .rtf ».

Par définition, une balise comporte au moins un caractère d'échappement, de préférence en dehors des caractères affichables figurant dans les 128 premières positions de la table de codification ASCII, tel que 0x1 (notation hexadécimale), 0x2, 0x80, ... (ce caractère contient à la fois une notion de type de balise et une notion de longueur de la balise). De façon optionnelle, elle peut comporter en outre un ou

plusieurs caractères, de préférence différents du zéro 0x0, qui est traditionnellement réservé à la fin d'une chaîne de caractères.

Pour répondre aux différents types de problèmes précités, on utilise quatre types de balises appelées respectivement :

- balises de mise en forme,
- balises de recherche avancée,
- balises de lancement de processus,
- balises de formatage ou d'alerte.

Pour simplifier la présentation, on a retenu ce découpage par catégorie, mais selon le type d'utilisation, on pourra faire appel à tel ou tel type de balise.

Balises de mise en forme.

Ces balises sont utilisées pour insérer des informations de mise en page. Par exemple pour afficher le mot «horizontal» on utilisera la séquence :

« h-o-0x8-G-r-i-z-0x8-S-o-0x8-g-n-t -0x8-s-a -l »,

dans laquelle:

- le caractère d'échappement « 0x8 » signifie « balise de début ou fin » avec une longueur de balise de 2 caractères (caractère d'échappement compris),
- le caractère suivant « G » correspond à « début de gras », « g » à « fin de gras », « S » à « début de souligné », « s » à « fin de souligné » (les caractères « - » ont été ajoutés pour faciliter la compréhension, mais ne figurent pas dans la chaîne de caractères de la table de représentation des documents stockés).

Des balises de ce type peuvent aussi être utilisées pour changer la police de caractères, la taille de la police, indenter des paragraphes, changer l'interligne, indiquer un changement de page, etc.

De la sorte, un ensemble de balises utilisant 2, 3 ou davantage de caractères, permet en partant d'un document MS Word ou Acrobat Reader Pdf, de créer une séquence de caractères qui permet à la fois :

- un balayage rapide, comme cela est précisé ci-après,
- la génération d'un fichier en format « rtf » sensiblement équivalent au document de départ, ce qui évite dans la majorité des cas de conserver à la fois la table des aperçus et le fichier MS Word de départ.

On notera que MS Word, Visual C++, WinSdk, MSN, rtf sont des formats et marques déposées par Microsoft Inc. Acrobat Reader Pdf est une marque déposée par Adobe Inc.

Balises de recherche avancée.

1) Utilisation de balises pour l'accentuation.

Il est utile de pouvoir effectuer une recherche sur un mot en tenant compte des accents. Par exemple, si on lance une recherche avec le mot « andré », il est utile de pouvoir retrouver les documents qui contiennent le mot sans accent, par exemple une adresse de courrier électronique telle que « andre.dupont@xxx.com », ou bien avec une faute d'orthographe : «andrè».

On peut coder cette information de la manière suivante :

« a-n-d-r-é-0x7-e-0x7-è »,

la balise « 0x7 » signifiant que le caractère qui suit (« e » ou « è ») est équivalent au précédent (« é »).

2) Utilisation de balises pour répéter n fois le même caractère.

Il peut être également utile de comparer 2 chaînes de caractères comportant des espaces, comme dans l'exemple suivant :

« moteur de recherche » et « moteur de recherche ».

On peut résoudre le problème avec des balises de la manière suivante : d'abord, dans la chaîne à rechercher, on remplace les séquences d'espaces, par un seul espace ou mieux par le caractère non affichable 0x1, et dans la chaîne à balayer, on effectue la conversion suivante :

- pour les séquences d'espaces inférieures à 6 caractères, on utilise des balises utilisant un seul caractère, à savoir 0x1, 0x2, 0x3, 0x4, 0x5 (sans autre caractère à la suite) ce qui permet avec un seul caractère de résoudre ce problème très fréquent quand un texte est affiché avec la justification à droite et à gauche.
- Pour les séquences plus longues, on peut utiliser une convention classique telle que : 0x6 - longueur de la séquence - caractère répété.

3) Utilisation de balises pour accélérer l'analyse de contenu.

Quand on veut analyser un texte, il faut commencer par faire un certain nombre d'opérations du type analyse grammaticale, et mémoriser le résultat de cette analyse avec des balises, afin d'obtenir des verbes à l'infinitif, des noms au singulier, des articles, des conjonctions, etc.

Par exemple : « le printemps est chaud et sec» peut être codé :

«0x1-l-e»	0x1 = article
«0x2-p-r-i-n-t-e-m-p-s»	0x2 = nom commun singulier
«0x4-P-3-ê-t-r-e»	0x4-P-3 = verbe Présent 3ème personne
«0x7-c-h-a-u-d»	0x7 = adjectif singulier
«0x8-e-t»	0x8 = conjonction

Dans la mesure où le programme de balayage d'une table peut être rendu extrêmement rapide comme on le verra plus loin, on peut utiliser une table dite « table dictionnaire », ou un ensemble de tables contenant tous les mots possibles dans un langue donnée pour vérifier que chaque mot d'un document existe, et effectuer son analyse grammaticale.

Une telle table dictionnaire comporterait une séquence de blocs comportant un ou deux éléments selon la complexité du mot à analyser. Par exemple :

«0x1-l-e»	0x1 = article
«0x2-p-r-i-n-t-e-m-p-s»	0x2 = nom commun singulier
«c-h-e-v-a-u-x-0x3-c-h-e-v-a l»	0x3 = nom commun pluriel
«e-s-t-0x4-P-3-ê-t-r-e»	0x4-P-3 = verbe Présent 3ème pers.
«0x7-c-h-a-u-d»	0x7 = adjectif singulier

Pour les verbes réguliers on peut avoir :

- soit toutes les formes possibles de conjugaisons, comme «i-n-v-e-n-t-e-r-a-s-0x4-F-2--i-n-v-e-n-t-e-r», futur à la 2eme personne,
- soit une forme plus compacte associée à une règle de conjugaison, comme «i-n-v-e-n-t-0x5-R-1--i-n-v-e-n-t-e-r», verbe régulier du premier groupe.

De la sorte, la table de représentation sera enrichie avec des balises et des mots permettant d'effectuer plus facilement les autres opérations d'analyse de contenu, cet enrichissement pouvant s'effectuer au moment de la création d'un élément de la table de représentation, ou bien au moment de la création d'une « table secondaire des représentations ».

Par ailleurs, quand on veut analyser le contenu d'un document de type texte, l'ordre des mots est important, comme dans l'exemple « location de voiture » ou « voiture de location ». Ceci nécessite parfois de balayer le texte plusieurs fois.

Plutôt que de relancer le balayage à partir d'une adresse que l'on aura préalablement stockée, une autre solution, comme on l'a vu plus haut, consiste à créer une table secondaire de représentation et à dupliquer le document. Pour

faciliter l'analyse, il peut être judicieux, au moment de la duplication, d'insérer des balises analogues à celles décrites ci-dessus pour faciliter l'analyse du contenu.

On peut également imaginer un système où l'on génère tout un ensemble de tables secondaires de représentation, soit pour un document, soit pour un ensemble de documents qui contiennent une chaîne de caractères prédéterminée ou des balises d'un type donné.

4) Utilisation de balises pour des méta-données.

Les moteurs de recherche sur Internet en général procèdent de la manière suivante.

Quand un nouveau document doit être ajouté à une base de données, on commence par analyser son contenu en utilisant différentes techniques, dont une consiste à effectuer l'analyse grammaticale, comme décrit ci-dessus ; ensuite le résultat de cette analyse consiste à créer une liste de mots-clefs ou méta-données attachées à ce document. Ce sont ces méta-données qui sont placées dans ce que l'on appelle communément une liste inverse, et qui sont recherchées quand un utilisateur fournit plusieurs critères pour rechercher un document.

Une méta-donnée de ce type peut-être codée au moyen d'un système de balise comme dans les exemples ci-dessous :

«0x14-2-3-é-t-a-l-o-n».

La balise 0x14 et les 2 caractères suivants (2-3) permettent de désigner le mot et de lui associer une notion telle que « 23 = animal ».

«0x15-1-3-r-e-f-i-n-a-n-c-e-m-e-n-t-0x15-f-i-n-a-n-c-e-r».

La balise 0x15 est d'une nature voisine et permet en plus d'associer une notion telle que l'action de financer.

De la sorte lors de la création initiale, ou bien par la suite lors de la création d'une « table secondaire des représentations » il est possible d'ajouter à un document toute une série de méta-données pour permettre une recherche intelligente sur le contenu.

5) Utilisation de balises pour l'écriture phonétique.

Si on veut interfacer la recherche avec un module de reconnaissance vocale, ou pour faciliter l'analyse automatique, il est utile de recourir à la phonétique. Dans une langue donnée, il y a en général une équivalence entre les mots et la façon de les prononcer, mais ce n'est pas toujours le cas comme le mot « parent » selon qu'il s'agit du «père» ou du verbe «parer». De la même manière, au même sons peuvent

être associées plusieurs orthographes particulièrement avec les noms propres comme « Durand » et « Durant ». Pour lever ce dilemme, après chaque mot qui pose un problème on peut placer une balise pour indiquer l'équivalent en écriture phonétique.

6) Utilisation de balises pour les montants.

Selon la langue, 1000 unités monétaires s'écrit de manière différente : en français, «1.000,00», ou «1.000», en anglais «1,000.00», etc.

Selon que l'utilisateur est français ou américain, il lancera sa recherche avec « 1.000,00 » ou « 1,000.00 » ou tout simplement « 1000 ». On peut utiliser un système de balises qui tient compte de cette particularité :

« 0x3-1-0-0-0-0-0-0x4-1-.-0-0-0-,-0-0-0x5-1-,-0-0-0-.-0-0-0x6 ».

La balise 0x3 indique que le champ suivant est un montant exprimé en centimes.

La balise 0x4 indique que le champ suivant est un montant affiché avec les conventions européennes.

La balise 0x5 indique que le champ suivant est un montant affiché avec les conventions américaines.

La balise 0x6 indique la fin de la zone relative à ce montant.

On peut aussi ajouter une balise pour indiquer quelle convention est utilisée dans le document de départ.

Ce système de balises permet de restituer la formulation de départ dans le document, et de retrouver ce montant quel que soit l'utilisateur qui lance une recherche.

7) Utilisation de balises pour les dates et heures.

On résout de façon analogue le problème des dates et des heures qui sont affichées de multiples manières selon la langue, le fuseau horaire, le fait d'afficher sans l'heure, etc.

8) Utilisation de balises pour les nombres.

D'une manière analogue, on peut utiliser une balise telle que 0x1C pour signifier que les quatre caractères suivants correspondent à un nombre entier codé en binaire sur 32 bits. Dans ce cas, la zone à comparer ne sera pas une chaîne de caractères, mais un nombre entier codé sur 32 bits. Il faut noter que dans ce cas précis, chacun des quatre caractères qui suivent la balise peut prendre une valeur quelconque, y compris le zéro binaire qui habituellement signale la fin d'une chaîne de caractères.

Ce mode de codification peut être utilisé pour tout type d'information numérique, signée ou non, sur 16, 64, 128 bits, en virgule flottante, etc. La comparaison entre deux zones pourra consister à tester l'égalité entre ces deux zones, mais d'une manière générale, on pourra effectuer toutes les opérations logiques entre deux zones numériques (plus petit, plus grand, ou logique, ou exclusif, etc.).

Il faut noter également que s'agissant de montants, selon les cas, on mémorisera l'information :

- soit sous forme plutôt texte, comme expliqué plus haut.
- soit sous forme plutôt numérique, c'est-à-dire :
  - une balise indiquant une monnaie (dollar, euro, ou autre),
  - une balise précisant la convention d'affichage (européenne ou anglo-saxonne),
  - une balise précédant un entier codé sur 32 bits,
  - enfin un nombre exprimant le montant en centimes.

Il va de soi que pour les cas les plus fréquents, une seule balise peut remplacer les 3 balises décrites ci-dessus.

Dans le cas où l'information est sous forme dite numérique, il faudra commencer par convertir la requête de l'utilisateur d'un format texte vers un format numérique, de façon à pouvoir effectuer la comparaison à grande vitesse, caractère par caractère.

Un montant est une zone dite de type numérique, mais il y en a d'autres. Ainsi, il en est de même pour les dates qui peuvent être mémorisées soit sous forme de texte, soit sous forme d'un nombre, selon les conventions couramment utilisées en informatique. Des balises peuvent préciser le mode d'affichage, le fait qu'il s'agisse d'une date exprimée en heure locale, ou mieux en temps universel.

Balises de lancement de processus.

1) Utilisation de balises pour déclencher un processus d'analyse.

Dans un document, il y a des mots qui ont signification plus importante que d'autres si on veut effectuer une analyse de son contenu. On peut faire ressortir ces mots par un système de balises du type :

« 0x16-2-3-f-a-i-l-l-i-t-e-0x16 »,

la balise 0x16 et les 2 caractères suivants (2-3) permettant à la fois de désigner le mot et de lui associer une notion telle que « 23 = juridique ».

Une corrélation entre les critères fournis par l'utilisateur et la présence de certains mots dans le document peut activer un processus d'analyse du contenu.

2) Utilisation de balises pour lancer d'autres programmes.

Par exemple si on veut protéger une information sensible, on peut utiliser une balise telle que :

« 0x17-p-a-s-s-w-o-r-d-1-0x17 »,

la balise 0x17 encadrant l'appel à une authentification de type 1, selon le résultat de laquelle le bloc d'informations en cours est ignoré ou analysé.

D'une manière générale, il s'agit d'un moyen de lancer une séquence d'instructions qui sont exécutées dans le même programme, ou bien dans un autre programme résidant sur la même machine ou sur une machine distante, permettant un mode de travail soit coopératif, soit en parallèle, selon les techniques habituelles de programmation.

Balises de Formatage ou d'alerte.

On peut considérer qu'une chaîne de caractères peut contenir à la fois un texte à afficher, des informations pour afficher celui-ci avec une présentation voisine de celle offerte par les outils de traitement de texte, des élément pour faciliter la recherche, des informations pour lancer des programmes.

Certains mots repérés par des balises, peuvent être saisis à la volée, et dupliqués dans une zone de mémoire en vue d'un traitement ultérieur pour analyser le contenu et permettre une recherche plus pertinente.

D'une manière plus générale, on pourra utiliser des balises pour donner des significations particulières à certains champs, tels qu'un numéro de compte, une quantité, un montant, une date, un code d'article, un pointeur vers un objet, une notion de hiérarchie, de parent, enfant, frère, c'est-à-dire toutes les notions que l'on peut trouver dans une table ou un fichier dans un ordinateur contenant une succession d'enregistrements de type différents. Par « enregistrement », on entend ici document stocké dans l'ordinateur.

On peut utiliser tout un jeu de balises pour un enregistrement tel qu'une opération bancaire, puis utiliser des balises avec les mêmes valeurs exprimées en binaire, mais avec une signification complètement différente pour un enregistrement correspondant à un stock de marchandises.

Ainsi, chaque type d'enregistrement, c'est-à-dire chaque type de document stocké dans l'ordinateur, peut être associé à un jeu de balises à significations spécifiques.

Au cours d'une opération complexe, par exemple pour éditer un relevé de compte bancaire, faisant intervenir plusieurs informations telles que le nom et l'adresse du titulaire du compte bancaire, la liste de tous les mouvements d'une période, on peut être amené à consulter plusieurs tables différentes de représentation des documents stockés, et la signification des balises pourra changer au cours des différentes phases de cette opération.

Une façon de résoudre le problème, est de mémoriser, soit au niveau de la table de représentation elle-même, soit au niveau de chaque enregistrement de la table de représentation, une information (ou un code) permettant de connaître la signification de tout le jeu de balises qui doit être utilisé à un moment donné.

On peut aussi utiliser une balise suivie d'une zone numérique sur 32 bits correspondant à une longueur L pour indiquer que les L caractères suivants correspondent à une zone sans texte, par exemple une image dans tel ou tel format, un son, une séquence d'image, une zone compressée ou codée en format « .zip », une séquence d'octets, un tableau de type MS Excel, et de manière générale une séquence de caractères sur laquelle on n'effectue pas de recherche.

On peut aussi utiliser des balises pour délimiter différentes zones de codage.

Dans le monde occidental, et particulièrement chez les anglo-saxons, la quasi-totalité de l'information affichable est codée sur un octet. Par contre pour des langues telles que l'arabe ou le chinois, ou pour quelques caractères tels que l'Euro, on utilise la notation Unicode.

En occident, on peut supposer que par défaut, le codage se fait sur un seul caractère, sauf entre une balise de début et une balise de fin de codage Unicode.

Dans le même esprit, sur 8 bits, c'est-à-dire un octet, on peut coder les 160 caractères de l'alphabet latin (10 chiffres, 2x26 lettres, 2x6x4 voyelles accentuées et environ 50 caractères spéciaux) et avoir une centaine de balises. La codification Unicode peut-être être remplacée par une autre codification plus compacte et mieux adaptée à cette utilisation.

S'il y a trop de combinaisons pour coder à la fois les caractères à afficher et les balises sur un seul caractère, c'est-à-dire plus de 256 possibilités pour un caractère sur 8 bits, on peut utiliser, pour les caractères les moins fréquents, par exemple les fractions, une balise indiquant que le caractère suivant appartient à un

deuxième jeu de caractères ; il faut noter que ce système est différent du système Unicode, qui lui utilise systématiquement 2 caractères, ce qui permet 65.536 possibilités, alors que le présent système ne permet que 256 caractères possibles derrière une balise de ce type.

Une table de représentation telle que décrite précédemment, c'est-à-dire incluant des balises, peut être utilisée de plusieurs manières :

- lancer une recherche à l'identique : on ignore tous les champs désignés par les balises : c'est par exemple un mode d'utilisation par défaut ;
- afficher un document dans une fenêtre d'aperçu, ou bien reconstituer le document original : pour cela, on ignorera toutes les balises, sauf celles de mise en forme ;
- lancer une recherche plus sophistiquée, avec une capacité d'interprétation du document : on utilisera toutes les balises de recherche avancée, y compris les balises de lancement de processus utiles pour mettre en œuvre les techniques connues les plus avancées dans ce domaine ;
- enfin, dans un domaine complètement différent, grâce à l'ensemble de ces techniques, utiliser cette table comme une véritable base de données avec des champs de toutes natures, des zones de type numérique, stockées sous forme décimale ou hexadécimale, des pointeurs, des zones pour lancer des processus, etc.

Toutes ces possibilités peuvent être regroupées en un petit jeu d'instructions appelées couramment API (de l'anglais « Application Program Interface »).

On trouvera ci-après un exemple d'une liste non limitative de ces API, à savoir :

- StrStrEx, par analogie avec la fonction « strstr » qui existe dans la plupart des langages de programmation, et qui consiste à rechercher dans une chaîne de caractères, la prochaine occurrence d'une sous chaîne donnée ;
- ExtractEdit, pour extraire d'une chaîne, le texte à éditer avec les seules balises relatives à la mise en page (le cas où on veut le texte brut sans aucune balise est un cas particulier de celui-ci) ;

- ExtractData, pour extraire les données d'une chaîne vers un ensemble de champs selon les formats utilisés habituellement en informatique (entier sur 32 bits ou 64 bits, format en virgule flottante, etc.) ;
- MakeEditStr, opération inverse de ExtractEdit pour convertir un ensemble de documents texte (tels que MS Word, rtf, etc., ou des courriers électroniques en format brut ou html) en une table de représentation avec des balises de mise en forme, et éventuellement celles permettant une recherche à partir de l'analyse du contenu ;
- MakeDataStr, opération inverse de ExtractData pour convertir chaque enregistrement d'un fichier en élément d'une table de représentation avec des balises permettant l'accès rapide à un élément au moyen de critères ;
- StrStrExMultiple, faisant appel plusieurs fois à la fonction élémentaire StrStrEx, et permettant de traiter plusieurs chaînes de caractères contenues dans un même document appelé document multiple afin d'y retrouver une ou plusieurs sous chaînes ;
- InitStrStrEx, pour définir la liste de toutes les balises, avec :
  - leur valeur (caractère d'échappement + premier caractère + deuxième caractère, ...),
  - leur signification et leur mode de fonctionnement dans les différents types d'utilisation (recherche, extraction pour édition, extraction pour conversion, lancement de traitements, ...), et d'une manière générale tous les éléments paramétrables ou ceux nécessaires pour relier les balises à des programmes externes.

Description de la fonction StrStrEx et mode de fonctionnement.

```
LPCSTR StrStrEx ( LPCSTR     ptrStart,
                  LPCSTR     ptrSubChain,
                  UINT       uiParameter,
                  STRSTREX   *strExtended)
```

dans laquelle :

LPCSTR	ptrStart	est le point départ dans la chaîne à explorer,
LPCSTR	ptrSubChain	la sous chaîne recherchée.
UINT	uiParameter	le mode de balayage,

STRSTREX *strExtended l'adresse d'une structure permettant de spécifier des données, des formats de conversion ou de communiquer avec d'autres processus.

Le mode de balayage est un ensemble de 32 bits ou plus qui, combinés, précisent comment on doit interpréter la chaîne de caractères. Par exemple :

- STREX_SKIP_BAL = -1 Ignorer la casse et toutes les balises,
- STREX_WITH_CASE = 1 Respecter la casse,
- STREX_SKIP_EDIT = 2 Ignorer les balises relatives à la mise en page,
- STREX_SKIP_ANALYSIS = 4 Ignorer les balises de recherche avancée,
- STREX_SKIP_PROCESS = 8 Ignorer les lancements de processus,
- STREX_SKIP_FORMAT = 16 Ignorer les balises de formatage,
- STREX_ FAST_DUPLIC = 32 Dupliquer certains mots à la volée,
- STREX_ ANALYSIS_1 = 64 Utiliser les balises de recherche avancée type 1,
- STREX_ ANALYSIS_2 = 128 Utiliser les balises de recherche avancée type 2,
- etc.

STRSTREX *strExtended est l'adresse d'une structure permettant de spécifier des données, des formats de conversion ou de communiquer avec d'autres processus, comme le fait la structure BROWSEINFO utilisée par l'API connue SHBrowseForFolder (cf. le WinSdk de Visual C++).

Par exemple, la commande « 0x17-p-a-s-s-w-o-r-d-1-0x17 » peut lancer un programme d'authentification désigné dans une commande de type «Callback».

La valeur retournée est :

- un pointeur sur la prochaine occurrence trouvée,
- 0x0 si aucune chaîne n'a été trouvée, ou
- une valeur symbolique en cas d'erreur.

Pour être performante, la fonction StrStrEx doit utiliser au mieux les caractéristiques des microprocesseurs modernes et les possibilités offertes par la technologie des composants électroniques. En particulier, il est exclus d'utiliser telles

quelles certaines fonctions fournies dans les bibliothèques du langage de programmation C.

On notera que l'objectif n'est pas d'avoir un code compact, mais d'exécuter le moins d'instructions possible pour les cas statistiquement les plus fréquents.

On trouvera en annexe un exemple de code écrit en langage C pour une partie de la fonction StrStrEx.

Description de la fonction ExtractEdit et mode de fonctionnement.

```
int ExtractEdit ( LPCSTR        ptrStart,
                  LPSTR         *ptrEditChain,
                  UINT          uiParameter
                  STRSTREX_ED   *strEditInfo)
```

dans laquelle :

LPCSTR	ptrStart	est l'adresse de la chaîne à extraire,
LPSTR	*ptrEditChain	l'adresse d'un pointeur sur la chaîne à éditer,
UINT	uiParameter	précise le mode d'édition (aucune mise en page, mise en page pour afficheur, mise en page pour restaurer un document MS Word en format rtf, etc.),
STRSTREX_ED	*strEditInfo	l'adresse d'une structure pour communiquer plus d'informations sur le mode de conversion et le format.

La fonction ExtractEdit utilise une grande partie des éléments de StrStrEx.

Description de la fonction ExtractData et mode de fonctionnement.

```
int ExtractData ( LPCSTR                ptrStart,
                  void                  *ptrExtractedData,
                  STRSTREX_EXTRACT      *strExtractInfo)
```

dans laquelle:

LPCSTR	ptrStart	est l'adresse de la chaîne à extraire,
LPSTR	*ptrExtractedData	l'adresse d'un pointeur sur l'objet à créer,
STRSTREX_ EXTRACT	*strExtractInfo	l'adresse d'une structure pour communiquer le format d'objet à fabriquer, et tous les traitements nécessaires pour effectuer la conversion.

La fonction ExtractData utilise une grande partie des éléments de StrStrEx.

Les fonctions MakeEditStr, et makeDataStr sont essentiellement des programmes de conversion qui ne posent pas de problème particulier pour un homme de l'art.

Description de la fonction StrStrExMultiple et mode de fonctionnement.

```
LPCSTR StrStrExMultiple ( LPCSTR          ptrStart,
                          LPCSTR          *ptrSubChain,
                          STRSTREX_MUL    *strExtended)
```

dans laquelle:

LPCSTR ptrStart est le point de départ dans la chaîne à explorer,

LPCSTR *ptrSubChain un ensemble de sous chaînes recherchées,

STRSTREX_MUL *strExtended l'adresse d'une structure permettant de spécifier les paramètres de cette fonction.

La valeur retournée est :

- un pointeur sur la prochaine occurrence trouvée,
- 0x0 si aucune chaîne n'a été trouvée, ou
- une valeur symbolique en cas d'erreur.

La fonction StrStrExMultiple permet de traiter le cas d'un document multiple tel qu'un courrier électronique.

Un courrier électronique regroupe des informations sur l'émetteur, les destinataires, les personnes en copie, l'objet, le contenu du courrier électronique, ainsi que d'autres informations, et ce courrier électronique se trouve stocké dans la table des aperçus sous la forme d'un en-tête, suivi des différentes chaînes émetteur, destinataires, personnes en copie, objet et contenu du courrier électronique, ledit en-tête comportant lui-même une balise de début, et lesdites autres informations.

En utilisant plusieurs fois la fonction élémentaire StrStrEx, il est possible de déterminer si une ou plusieurs chaînes du document multiple contiennent une sous chaîne recherchée, et dans quelle chaîne. Il est possible de déterminer également si le document multiple contient non pas une seule sous chaîne, mais plusieurs sous chaînes recherchées.

Description de la fonction InitStrStrEx et mode de fonctionnement.

```
int InitStrStrEx ( STRSTREX_BALISES      *strBalises,
                   STRSTREX_PROCESS      *strProcess,
                   STRSTREX_CONV_CHAR    *strConvChar,
                   STRSTREX_MISC         *strMisc)
```

dans laquelle:

STRSTREX_BALISES *strBalises est l'adresse d'une structure spécifiant les valeurs des échappements, de chacune des balises, leur longueur, catégorie (mise en page...) leur action, les liaisons avec les traitements, etc.,

STRSTREX_PROCESS *strProcess est l'adresse d'une structure spécifiant les informations pour effectuer la résolution des liens avec les traitements externes ou internes utilisés par les StrStrEx et les autres API décrites ci-dessus,

STRSTREX_CONV_CHAR *strConvChar est l'adresse d'une structure spécifiant la liste des caractères utilisés, Unicode, Ascii, etc, les tables de conversions entre ces codifications, les règles de passage de majuscule à minuscule, etc.,

STRSTREX_MISC *strMisc est l'adresse d'une structure spécifiant les autres données telles que version, langues, langages de programmation, système d'exploitation (Windows, Unix, Linux...), les conventions de codage (xml, rtf, MS Word, etc), les limites en vitesse de processeur, taille mémoire, taille des entiers, etc.

Cette fonction est en général lancée au début de toute exécution d'un programme utilisant l'API StrStrEx et ses dérivés.

Au moins une partie de ces fonctions peut être regroupée dans ce que l'on appelle une bibliothèque qui peut être intégrée dans d'autres applications.

Par exemple, cette bibliothèque peut être intégrée dans d'autres applications pour construire un moteur de recherche basé sur la technique de balayage d'une table de représentation telle que décrite précédemment, qui a la particularité de :

- pouvoir intégrer une fenêtre d'aperçu dont le contenu est extrait de ladite table, et
- grâce aux balises de mise en page, offrir en plus une présentation équivalente aux documents de départ dans la majorité des cas.

Cette bibliothèque peut également être intégrée dans d'autres applications pour construire ou analyser un conteneur regroupant à la fois :

- des documents comportant du texte tels que MS Word ou Pdf provenant du disque local ou du réseau local d'un utilisateur,

- des courriers électroniques avec leurs pièces jointes, c'est-à-dire des documents comportant du texte (MS Word, pdf, etc.) ou tout document tel que image, son, etc., et
- les éléments suffisants pour avoir un aperçu des documents comportant du texte, sans avoir à ouvrir ces documents avec le programme associé, ce que l'on obtient en insérant un des éléments de ladite table de représentation des documents stockés.

Grâce à une codification avec des balises de mise en page, il est possible de supprimer la plus grande partie des documents de type texte tels que MS Word ou Pdf puisque ladite table contient le plus souvent une information équivalente.

Il faut savoir que les documents Pdf et surtout MS Word sont en général 10 fois plus volumineux qu'un document en format rtf équivalent et a fortiori qu'un fichier utilisant des balises très compactes comme celui qui est décrit ci-dessus.

Un tel gain de place est fort utile, à la fois pour sauver l'information sur disque, pour générer des sauvegardes, pour constituer des archives pour les courriers électroniques, pour transporter ces informations sur les réseaux locaux ou via Internet sous forme de pièces jointes dans les courriers électroniques. Ceci permet d'éviter que beaucoup d'utilisateurs de grandes entreprises soient obligés de supprimer leurs courriers électroniques vieux de plus de 6 ou 12 mois, ce qui constitue une gêne importante pour eux.

Cette bibliothèque peut également être intégrée dans d'autres applications pour construire les différents éléments d'un logiciel de messagerie pour :

- intégrer un moteur de recherche avec les caractéristiques décrites ci-dessus, et
- offrir un nouveau système de pièce jointes utilisant un conteneur décrit ci-dessus.

Cette bibliothèque peut également être intégrée dans d'autres applications pour construire des bases de données contenant essentiellement des informations non modifiables comme on le voit dans l'exemple ci-dessous.

Une banque comporte un million de clients, et l'ensemble des courriers électroniques y compris les pièces jointes, des courriers ou documents spécifiques d'un client représente en moyenne vingt mille caractères (soit environ dix pages pleines). L'ensemble de ces données, avec les balises pour la mise en page, plus les identifiants (code agence, numéro de compte, dates, textes spécifiques, références

des divers courriers, adresses électroniques, etc.) et les balises de formatage correspondantes, représente au maximum 32 Ko.

Un client compte en moyenne une vingtaine de mouvements par mois, et il faut en moyenne environ une centaine de caractères pour décrire un mouvement comptable : code agence, code opération, numéro de compte, dates, montant, texte associé tel que « virement à Monsieur Untel » ou « chèque No 12345 », numéro d'imprimé utilisé pour imprimer le relevé de compte.

L'ensemble des mouvements d'un client pendant une année, avec les balises correspondantes représente au maximum 32 Ko.

L'ensemble de toutes ces informations non modifiables, à savoir tous les documents de type texte dans la vie d'un client ainsi que tous les mouvements comptables pendant un an représente 64 Go, qui pourraient facilement tenir dans le disque dur d'un simple micro-ordinateur.

Quand il y a un nouveau document, ou un nouveau mouvement comptable, il suffit de le rajouter à la fin de la table de représentation des documents stockés, ce qui rend inutile l'utilisation de pointeurs ou de tables de correspondance en tout genre qui posent des problèmes de mise à jour, et surtout de reprise en cas d'incident, pour une simple question de cohérence entre les différentes informations.

Si l'on veut afficher tous les mouvements comptables d'un client pendant les quinze derniers jours à partir d'un poste de travail dans une agence, on procédera comme suit :

- à partir du poste de travail, on lance une requête vers une base de données distante pour rechercher toutes les opérations correspondant à un numéro de compte donné, entre deux dates prédéterminées.
- en retour, tous les mouvements, avec les contenus et les balises, tels que décrits ci-dessus, sont renvoyés par le réseau interne de la banque, de la base de données vers le poste de travail, et peuvent être affichés à l'écran.

Si l'on veut imprimer un relevé de compte grâce au numéro d'imprimé utilisé pour imprimer le relevé de compte, il sera possible d'imprimer un relevé ce compte identique à celui qui avait été envoyé au client. Comme on peut le constater la fonction ExtractData peut utilement être déportée dans une machine autre que celle qui contient la base de données.

Un des principaux intérêts de ce procédé, c'est que c'est la même séquence de caractères qui figure dans la base de données, et qui est utilisée en fin de

traitement pour imprimer le document, et cette chaîne de caractères est très compacte, ce qui a pour effet de réduire le trafic sur le réseau.

Pour obtenir des temps d'accès compatibles avec les applications évoqués ci-dessus, il y a plusieurs possibilités qui peuvent être mises en œuvre indépendamment les unes des autres, ou bien ensemble, le but étant toujours d'exécuter le plus rapidement possible la fonction StrStrEx, et en particulier la séquence d'instructions qui permet d'ignorer les caractères sans intérêt comme dans l'exemple ci-après : si on recherche la sous chaîne « information », il faut parcourir le plus rapidement possible la chaîne, en ignorant les balises de mise en page, jusqu'au moment où l'on rencontre un « i » majuscule ou minuscule, et quand on en a trouvé un, déterminer rapidement si le caractère utile suivant est un « n » majuscule ou minuscule.

Parmi ces différentes possibilités, on peut citer : optimiser le code en langage assembleur, utiliser des microprocesseurs performants pour exécuter ce type de programme en raison de la taille de la mémoire cache ou de leur aptitude à exécuter plusieurs instructions en un seul cycle d'horloge, utiliser des processeurs travaillant sur 64 bits voire plus.

Comme cela est représenté sur la figure 5, on peut utiliser plusieurs microprocesseurs Co-Pi ou ordinateurs en parallèle travaillant chacun sur une partie MEMi de la table de représentation des documents stockés. Par exemple, on peut adjoindre à un simple micro-ordinateur avec 4 Go de mémoire, une carte du type DSP32 équipée de 16 microprocesseurs travaillant chacun en parallèle sur 1/16ème de la table de représentation complète.

On encore utiliser un microprocesseur supportant la technologie FPGA (de l'anglais « Field Programmable Gate Array ») et créer la succession de portes logiques correspondant à la partie de la fonction StrStrEx qui doit être exécutée très rapidement.

Une autre possibilité est d'utiliser un microprocesseur qui est capable, en quelques cycles d'horloge, d'exécuter une séquence de plusieurs dizaines, ou centaines, ou milliers d'instructions qui ne sont pas stockées dans la mémoire de la machine, et chargées à chaque fois dans la mémoire cache du microprocesseur, mais gravées au moins en partie dans le microprocesseur lui-même, à la manière des composants spécialisés comme les processeurs graphiques qui permettent l'affichage rapide d'une image haute définition.

Selon les cas, au moins une partie de la bibliothèque des API peut, soit être ajoutées à un microprocesseur existant, ce qui permet d'obtenir un balayage rapide avec un simple micro-ordinateur, par exemple pour effectuer des recherches dans des courriers électroniques, soit être placées dans un microprocesseur séparé, appelé co-processeur Co-Pi, qui accède à la mémoire de la machine, et exécute ses instructions sous le contrôle d'un autre microprocesseur maître MainProc, comme le fait le processeur graphique d'un micro-ordinateur (cf. figure 5).

Utilement on peut également placer dans le microprocesseur une ou plusieurs tables dictionnaire, en vue d'accélérer l'analyse grammaticale d'un document.

ANNEXE

Exemple de code écrit en langage C pour une partie de la fonction StrStrEx

```
/*****
dans l'exemple suivant, on a supposé que l'on recherche la chaîne de caractères "lévrier"
– tous les caractères affichables sont compris entre 0x1 et BALISE_MINI -1 ;
– toutes les balises sont comprises entre les valeurs BALISE_MINI et BALISE_MAXI, à savoir :
    ▪ BALISE_MINI2 et BALISE__MAXI2 pour les balises à 2 caractères,
    ▪ BALISE_MINI3 et BALISE__MAXI3 pour les balises à 3 caractères,
    ▪ BALISE_SAME_CHAR est la balise pour substituer un caractère (pour trouver "lévrier" ou "levrier"),
    ▪ etc.
*****/

LPCSTR StrStrEx (LPCSTR ptrStart, LPCSTR ptrSubChain, UINT uiParameter, STRSTREX *strConvFormat)
 {
  strupr (ptrSubChain) ;
  BYTE  ucFirstCharUpr      = ptrSubChain [0] ;
  BYTE  ucSecondCharUpr     = ptrSubChain [1] ;

  strlwr (ptrSubChain) ;
  BYTE  ucFirstCharLwr      = ptrSubChain [0] ;
  BYTE  ucSecondCharLwr     = ptrSubChain [1] ;

  // boucle très courte pour traiter d'abord les cas statistiquement les plus fréquents,
  // l'objectif n'est pas d'avoir un code compact, mais rapide
  while ( TRUE)
   {
    if (          *ptr == 0)
      break ;
```

```
if (            *ptr < BALISE_MINI
  &&            *ptr != ucFirstCharLwr
  &&            *ptr != ucFirstCharUpr)
{
 ptr++ ;
 continue ;   // --> caractère suivant
}

if (            *ptr == BALISE_SAME_CHAR)
{
 ptr++ ;        // avancer d'un caractère pour tester le caractère suivant
 if (           *ptr != ucFirstCharLwr
   &&           *ptr != ucFirstCharUpr)
 {
  ptr++ ;
  continue ; // --> caractère suivant
 }
}
else if (*ptr <= BALISE_MAXI2 )
{
 ptr+= 2 ;    // avancer de 2 caractères pour tester le caractère suivant
 continue ;
}
else if (*ptr <= BALISE_MAXI3 )
{
 ptr+= 3 ;    // avancer de 2 caractères pour tester le caractère suivant
 continue ;
}

// ---- Ici, on a trouvé le premier caractère de la sous-chaîne ( 'l' de "lévrier") ------

 if (ucSecondCharLwr != 0) // protection si la sous-chaine comporte plus d'un
caractère
{
```

```
ptr++ ;

if (        *ptr == 0)
 break ;

if (        *ptr < BALISE_MINI
  &&        *ptr != ucSecondCharLwr
  &&        *ptr != ucSecondCharUpr)
 {
 ptr++ ;
 continue ;   // --> caractère suivant
 }

else if ( *ptr == BALISE_SAME_CHAR)
 {
 ptr++ ;      // avancer d'un caractère pour tester le caractère suivant
 if (        *ptr != ucSecondCharLwr
   &&        *ptr != ucSecondCharUpr)
  {
  ptr++ ;
  continue ; // --> caractère suivant
  }
 }
else if (*ptr <= BALISE_MAXI2 )
 {
 ptr+= 2 ;  // avancer de 2 caractères pour tester le caractère suivant
 continue ;
 }
else if (*ptr <= BALISE_MAXI3 )
 {
 ptr+= 3 ;  // avancer de 2 caractères pour tester le caractère suivant
 continue ;
 }
}
```

// ---- Ici, on a trouvé les premiers caractère de la sous-chaîne ('l' de "lévrier") --- // on peut effectuer la même opération pour Le 3ème caractère, ou bien effectuer une boucle.

REVENDICATIONS

1. Procédé de recherche d'informations dans des documents stockés dans une mémoire électronique, comportant les étapes suivantes :
   - sélection d'au moins un document parmi les documents stockés, à partir d'une requête comportant au moins une chaîne de caractères prédéterminée, puis
   - extraction d'un résultat en vue de son affichage sous forme d'un aperçu d'informations relatives au document sélectionné,
   - préalablement aux étapes de sélection et d'extraction, génération d'une table de représentation des documents stockés, comportant une chaîne de caractères comprenant au moins une partie des informations des documents stockés,

   caractérisé en ce que, lors de l'étape d'extraction, on génère le résultat à l'aide de la table de représentation, à partir d'informations contenues dans la chaîne de caractères de la table de représentation jugées pertinentes en fonction de la requête.
2. Procédé de recherche d'informations selon la revendication 1, dans lequel, lors de l'étape de sélection, on compare la chaîne de caractères prédéterminée de la requête à la chaîne de caractères de la table de représentation, notamment par balayage séquentiel de la table de représentation, pour sélectionner au moins un document parmi les documents stockés.
3. Procédé de recherche d'informations selon la revendication 1 ou 2, dans lequel, au moins un document stocké étant de type courrier électronique et comportant plusieurs rubriques distinctes choisies parmi l'ensemble d'éléments constitué d'une adresse d'un émetteur, d'une adresse d'un destinataire, d'un en-tête, d'un corps de message, et d'au moins une pièce jointe, la chaîne de caractères de la table de représentation comporte au moins une partie des informations de type texte de chaque rubrique du document de type courrier électronique.
4. Procédé de recherche d'informations selon les revendications 2 et 3, dans lequel pour le document de type courrier électronique, on balaye séquentiellement les informations concernant la pièce jointe avant les informations concernant toute autre rubrique de ce document.

5. Procédé de recherche d'informations selon l'une quelconque des revendications 1 à 4, dans lequel la chaîne de caractère de la table de représentation comporte en outre pour chaque document stocké des informations d'identification de ce document.

6. Procédé de recherche d'informations selon l'une quelconque des revendications 1 à 5, dans lequel on stocke en mémoire au moins une partie du résultat de la recherche d'informations.

7. Procédé de recherche d'informations selon l'une quelconque des revendications 1 à 6, dans lequel la partie du résultat de la recherche d'informations stockée en mémoire est stockée dans un fichier apte à comporter plusieurs résultats de plusieurs recherches.

8. Procédé de recherche d'informations selon l'une quelconque des revendications 1 à 7, comportant, lors de l'étape d'extraction du résultat, les étapes suivantes :
   - extraction des informations contenues dans la chaîne de caractères de la table de représentation jugées pertinentes en fonction de la requête,
   - transmission de ces informations vers un terminal distant par l'intermédiaire d'un réseau de transmission de données,

   et dans lequel l'affichage du résultat est réalisé par le terminal distant.

9. Procédé de recherche d'informations selon l'une quelconque des revendications 1 à 8, dans lequel, lors de l'étape de génération de la table de représentation des documents stockés, on effectue une conversion pour que tout caractère affichable d'une zone de type texte des documents stockés soit codé :
   - soit sur un octet ;
   - soit à l'aide d'une balise insérée dans la table de représentation et suivie d'un code sur un octet.

10. Procédé de recherche d'informations selon l'une quelconque des revendications 1 à 9, dans lequel, lors de l'étape de génération de la table de représentation, on insère dans la chaîne de caractères de la table de représentation au moins un ensemble de données délimité par au moins une balise pour compléter les informations comprises dans cette chaîne de caractères.

11. Procédé de recherche d'informations selon la revendication 10, dans lequel chaque balise insérée dans la chaîne de caractères comporte au moins un

caractère d'échappement codé sur un octet n'appartenant pas aux caractères affichables figurant dans les 128 premières positions de la table de codification ASCII.

12. Procédé de recherche d'informations selon la revendication 10 ou 11, dans lequel l'ensemble de données comporte des données d'aide à la présentation de l'aperçu, utilisées lors de l'étape d'extraction du résultat.

13. Procédé de recherche d'informations selon l'une quelconque des revendications 10 à 12, dans lequel l'ensemble de données comporte des données d'aide à la sélection d'au moins un document.

14. Procédé de recherche d'informations selon l'une quelconque des revendications 10 à 13, dans lequel on insère dans la chaîne de caractères de la table de représentation au moins une zone d'informations de type numérique codée sur un nombre prédéterminé d'octets délimité par au moins une balise d'indication de cette zone numérique.

15. Procédé de recherche d'informations selon la revendication 14, dans lequel la balise d'indication de la zone numérique est en outre une balise d'indication d'une convention de présentation de cette zone numérique.

16. Procédé de recherche d'informations selon l'une quelconque des revendications 10 à 15, dans lequel les documents stockés étant répartis en différents types de documents, on définit pour chaque type de documents un ensemble de balises destinées à être insérées dans la chaîne de caractères de la table de représentation, chaque balise de cet ensemble ayant une signification spécifique à ce type de documents.

17. Procédé de recherche d'informations selon l'une quelconque des revendications 10 à 16, dans lequel on insère dans la chaîne de caractères de la table de représentation au moins un ensemble de données exprimées en écriture phonétique délimité par au moins une balise d'indication d'écriture phonétique.

18. Procédé de recherche d'informations selon l'une quelconque des revendications 10 à 17, dans lequel on insère dans la chaîne de caractères de la table de représentation au moins une balise d'indication qu'un nombre prédéterminé de caractères suivant cette balise dans la chaîne de caractères de la table de représentation n'a pas à être balayé lors de l'étape de sélection.

19. Procédé de recherche d'informations selon l'une quelconque des revendications 10 à 18, dans lequel on insère dans la chaîne de caractères de

la table de représentation au moins un ensemble de données correspondant à une analyse grammaticale d'une partie du contenu d'au moins un document stocké, délimité par au moins une balise d'indication d'analyse grammaticale.

20. Procédé de recherche d'informations selon l'une quelconque des revendications 10 à 19, dans lequel on insère dans la chaîne de caractères de la table de représentation au moins un ensemble de données correspondant à des méta-données de description d'une partie du contenu d'au moins un document stocké, délimité par au moins une balise d'indication de méta-données.

21. Procédé de recherche d'informations selon l'une quelconque des revendications 10 à 20, dans lequel on insère dans la chaîne de caractères de la table de représentation au moins une balise pour lancer un programme prédéterminé.

22. Procédé de recherche d'informations selon l'une quelconque des revendications 1 à 21, dans lequel :

- chaque document stocké comportant des informations réparties dans plusieurs rubriques distinctes prédéterminées communes à tous les documents stockés, le résultat est affiché sous la forme d'un aperçu comportant une zone d'aperçu pour chaque rubrique distincte commune et comportant une liste de documents initialement sélectionnés pour des informations qu'ils contiennent jugées pertinentes en fonction de la requête,
- chaque zone d'aperçu est désactivable, et
- lorsqu'on désactive au moins une zone d'aperçu, on maintient uniquement dans la liste affichée chaque document initialement sélectionné pour des informations jugées pertinentes que ce document comporte dans au moins une rubrique correspondant à au moins une zone d'aperçu qui reste activée.

23. Moteur de recherche d'informations dans des documents stockés dans une mémoire électronique, comportant :

- des moyens de génération d'une table de représentation des documents stockés, cette table comportant une chaîne de caractères comprenant au moins une partie des informations des documents stockés,

– des moyens de sélection d'au moins un document parmi les documents stockés, à partir d'une requête comportant au moins une chaîne de caractères prédéterminée,

caractérisé en ce qu'il comporte des moyens d'extraction d'un résultat à l'aide de la table de représentation, à partir d'informations contenues dans la chaîne de caractères de la table de représentation jugées pertinentes en fonction de la requête, en vue de l'affichage de ce résultat sous forme d'un aperçu d'informations relatives au document sélectionné.

24. Microprocesseur comportant des instructions programmées pour la mise en œuvre d'un procédé de recherche d'informations selon l'une quelconque des revendications 1 à 22.

25. Microprocesseur selon la revendication 24, comportant en outre des moyens de stockage d'au moins une table dictionnaire comprenant un ensemble de mots dans une langue prédéterminée, chaque mot étant associé dans cette table dictionnaire à des données d'analyse grammaticale.

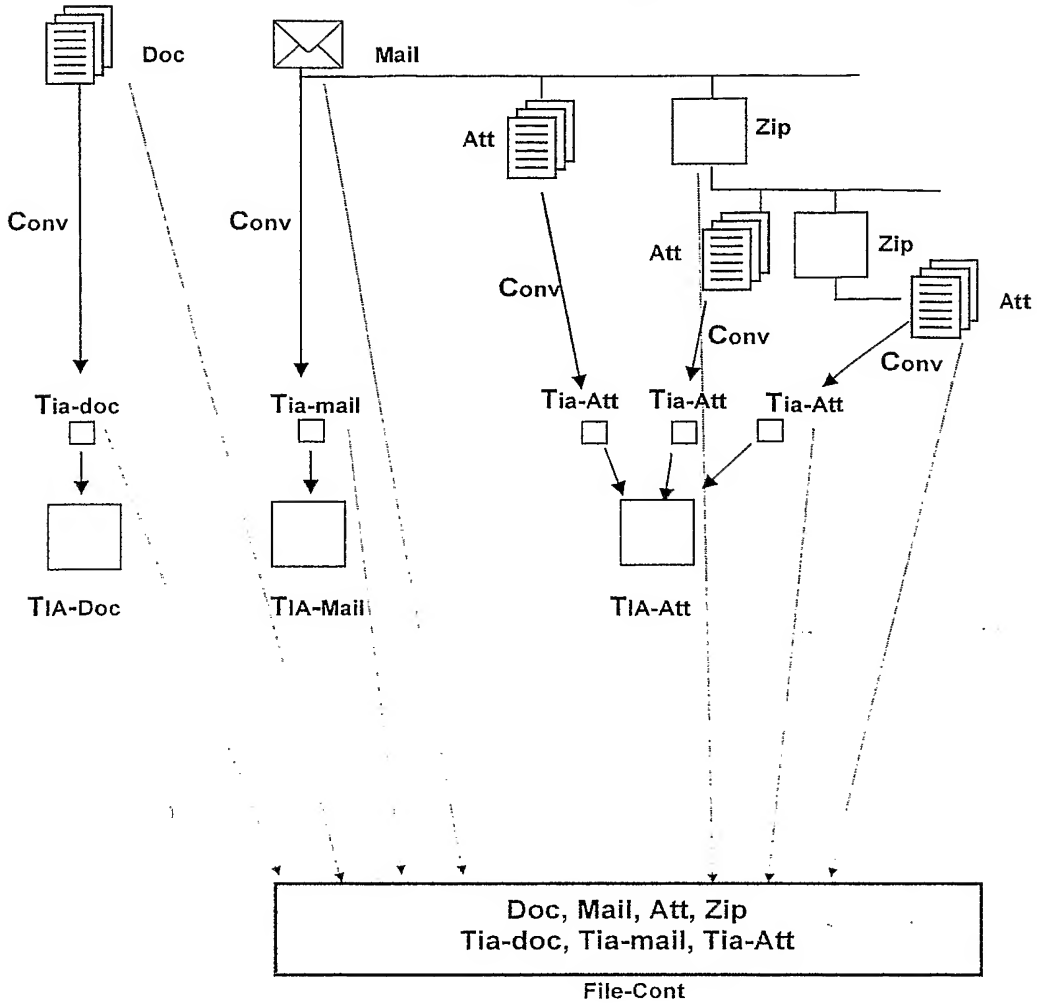

Figure 1

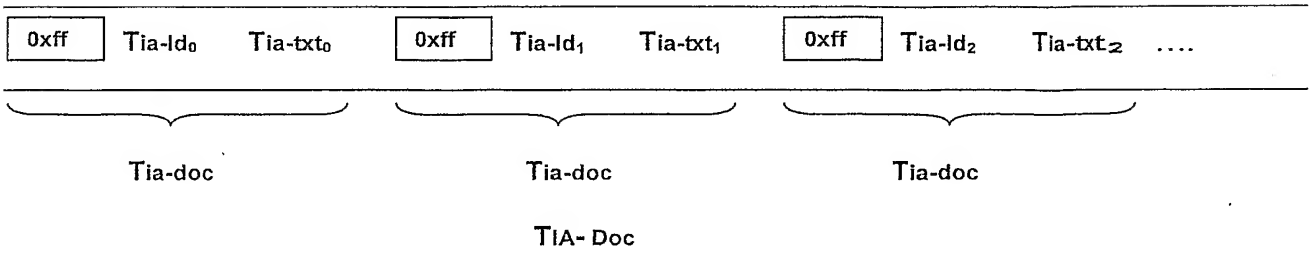

Figure 2

	Résultat de la recherche sur le mot « Paris »			
	De ☒	A ☒	Copie ☒	Objet ☒
Ligne **L1**	Paul Durand	Louis **Paris**		Moteur recherche
Ligne **L2**	Paul Durand	Jean-François Durand	**Paris**ot	Moteur recherche
Ligne **L3**	Paul Durand	**Paris**ot	Louis **Paris**	Moteur recherche
Ligne **L4**	Paul Durand	Louis **Paris**		Moteur recherche
	Colonne **C1**	Colonne **C2**	Colonne **C3**	Colonne **C4**

Figure 3

	Résultat de la recherche sur le mot « Paris »			
	De ☒	A ☒	Copie ☐	Objet ☒
Ligne **L1**	Paul Durand	Louis **Paris**		Moteur recherche
Ligne **L3**	Paul Durand	**Paris**ot	Louis **Paris**	Moteur recherche
Ligne **L4**	Paul Durand	Louis **Paris**		Moteur recherche
	Colonne **C1**	Colonne **C2**	Colonne **C3**	Colonne **C4**

Figure 4

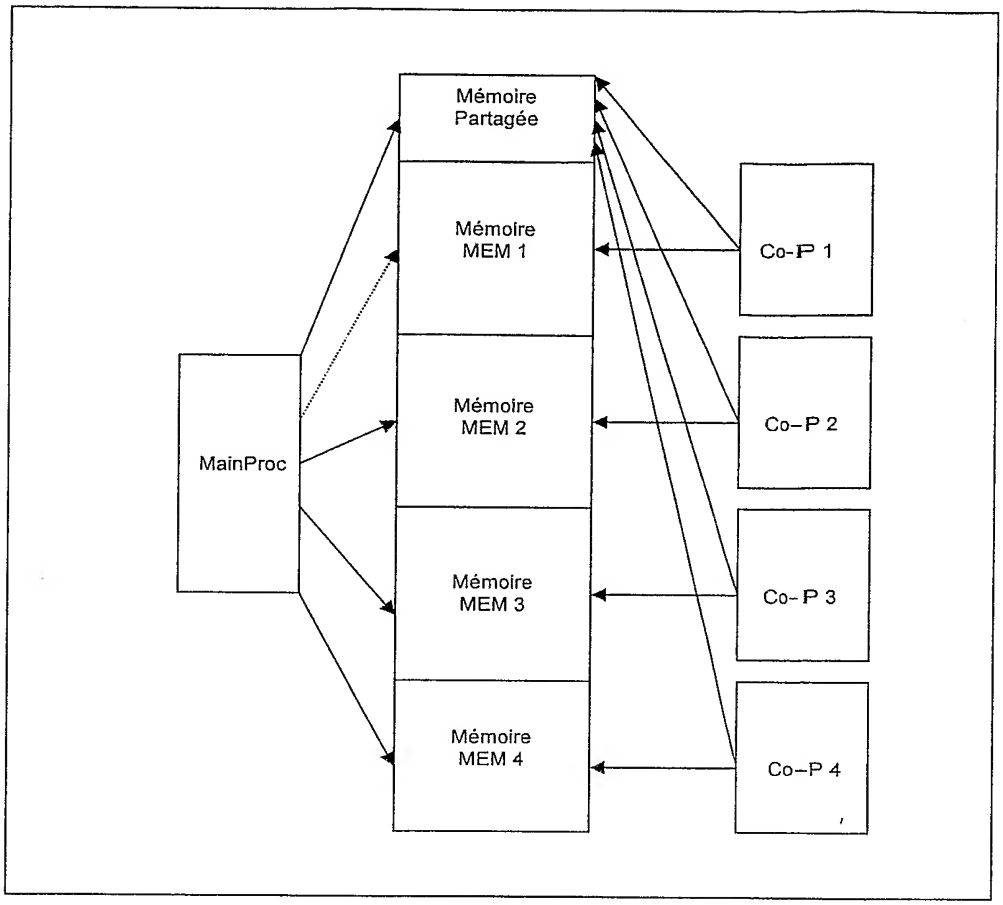
Mémoire Partagée
Mémoire MEM 1
Mémoire MEM 2
Mémoire MEM 3
Mémoire MEM 4
MainProc
Co-P 1
Co-P 2
Co-P 3
Co-P 4

Figure 5

INTERNATIONAL SEARCH REPORT

Int. Application No
PCT/FR2005/000659

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 02/065316 A (OTG SOFTWARE, INC) 22 August 2002 (2002-08-22) paragraph '0011! - paragraph '0012! paragraph '0020! - paragraph '0027! paragraph '0041! - paragraph '0045!; claims; figures	1—5, 8—15,20, 23,24
X	EP 0 886 227 A (DIGITAL EQUIPMENT CORPORATION; COMPAQ COMPUTER CORPORATION) 23 December 1998 (1998-12-23) column 6 - column 16; figures	1—3,5-8, 10-15, 20,23-25
P,X	US 6 721 748 B1 (KNIGHT TIMOTHY O ET AL) 13 April 2004 (2004-04-13) column 2 - column 6 column 10 - column 13	1,2,22

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 July 2005	29/07/2005

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Herry, T

INTERNATIONAL SEARCH REPORT

Inte l Application No
PCT/FR2005/000659

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2002/143871 A1 (MEYER DAVID FRANCIS ET AL) 3 October 2002 (2002-10-03) the whole document	3,4,9-12
P,A	EP 1 408 428 A (FRANCE TELECOM) 14 April 2004 (2004-04-14) claims	6,7,15, 16
A	EP 1 280 074 A (SCHNEIDER AUTOMATION) 29 January 2003 (2003-01-29) paragraph '0028! paragraph '0043! - paragraph '0056!	9-22
A	FR 2 715 486 A (PIATON ALAIN NICOLAS) 28 July 1995 (1995-07-28) page 7, line 31 - page 25, line 31	16-19

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte al Application No
PCT/FR2005/000659

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 02065316 A	22-08-2002	CA 2433525 A1	22-08-2002
		CN 1531688 A	22-09-2004
		EP 1368739 A1	10-12-2003
		WO 02065316 A1	22-08-2002
		US 2002122543 A1	05-09-2002
EP 0886227 A	23-12-1998	CA 2240556 A1	16-12-1998
		DE 69818549 D1	06-11-2003
		DE 69818549 T2	05-08-2004
		EP 0886227 A1	23-12-1998
		JP 11015759 A	22-01-1999
US 6721748 B1	13-04-2004	US 6493703 B1	10-12-2002
		US 6804675 B1	12-10-2004
US 2002143871 A1	03-10-2002	NONE	
EP 1408428 A	14-04-2004	FR 2845789 A1	16-04-2004
		EP 1408428 A1	14-04-2004
EP 1280074 A	29-01-2003	FR 2827686 A1	24-01-2003
		EP 1280074 A1	29-01-2003
		US 2003016242 A1	23-01-2003
FR 2715486 A	28-07-1995	FR 2715486 A1	28-07-1995
		WO 9520196 A1	27-07-1995
		US 5806073 A	08-09-1998

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale No
PCT/FR2005/000659

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
CIB 7 G06F17/30

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)
CIB 7 G06F

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si réalisable, termes de recherche utilisés)
EPO-Internal, INSPEC

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie °	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X	WO 02/065316 A (OTG SOFTWARE, INC) 22 août 2002 (2002-08-22) alinéa '0011! - alinéa '0012! alinéa '0020! - alinéa '0027! alinéa '0041! - alinéa '0045!; revendications; figures	1-5, 8-15,20, 23,24
X	EP 0 886 227 A (DIGITAL EQUIPMENT CORPORATION; COMPAQ COMPUTER CORPORATION) 23 décembre 1998 (1998-12-23) colonne 6 - colonne 16; figures	1-3,5-8, 10-15, 20,23-25
P,X	US 6 721 748 B1 (KNIGHT TIMOTHY O ET AL) 13 avril 2004 (2004-04-13) colonne 2 - colonne 6 colonne 10 - colonne 13	1,2,22

-/--

[X] Voir la suite du cadre C pour la fin de la liste des documents

[X] Les documents de familles de brevets sont indiqués en annexe

° Catégories spéciales de documents cités:

"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent
"E" document antérieur, mais publié à la date de dépôt international ou après cette date
"L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)
"O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens
"P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée
"T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention
"X" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément
"Y" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier
"&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée	Date d'expédition du présent rapport de recherche internationale
19 juillet 2005	29/07/2005

Nom et adresse postale de l'administration chargée de la recherche internationale	Fonctionnaire autorisé
Office Européen des Brevets, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Herry, T

RAPPORT DE RECHERCHE INTERNATIONALE

Der rnationale No
PCT/FR2005/000659

C.(suite) DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie °	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
A	US 2002/143871 A1 (MEYER DAVID FRANCIS ET AL) 3 octobre 2002 (2002-10-03) le document en entier	3,4,9-12
P,A	EP 1 408 428 A (FRANCE TELECOM) 14 avril 2004 (2004-04-14) revendications	6,7,15, 16
A	EP 1 280 074 A (SCHNEIDER AUTOMATION) 29 janvier 2003 (2003-01-29) alinéa '0028! alinéa '0043! - alinéa '0056!	9-22
A	FR 2 715 486 A (PIATON ALAIN NICOLAS) 28 juillet 1995 (1995-07-28) page 7, ligne 31 - page 25, ligne 31	16-19

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Demande internationale No
PCT/FR2005/000659

Document brevet cité au rapport de recherche		Date de publication	Membre(s) de la famille de brevet(s)		Date de publication
WO 02065316	A	22-08-2002	CA	2433525 A1	22-08-2002
			CN	1531688 A	22-09-2004
			EP	1368739 A1	10-12-2003
			WO	02065316 A1	22-08-2002
			US	2002122543 A1	05-09-2002
EP 0886227	A	23-12-1998	CA	2240556 A1	16-12-1998
			DE	69818549 D1	06-11-2003
			DE	69818549 T2	05-08-2004
			EP	0886227 A1	23-12-1998
			JP	11015759 A	22-01-1999
US 6721748	B1	13-04-2004	US	6493703 B1	10-12-2002
			US	6804675 B1	12-10-2004
US 2002143871	A1	03-10-2002	AUCUN		
EP 1408428	A	14-04-2004	FR	2845789 A1	16-04-2004
			EP	1408428 A1	14-04-2004
EP 1280074	A	29-01-2003	FR	2827686 A1	24-01-2003
			EP	1280074 A1	29-01-2003
			US	2003016242 A1	23-01-2003
FR 2715486	A	28-07-1995	FR	2715486 A1	28-07-1995
			WO	9520196 A1	27-07-1995
			US	5806073 A	08-09-1998